

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **08263333 A**(43) Date of publication of application: **11.10.96**

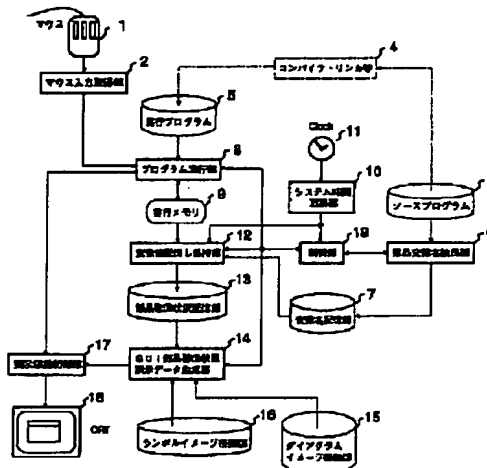
(51) Int. Cl

**G06F 11/34****G06F 9/06****G06F 9/06**(21) Application number: **07067925**(71) Applicant: **TOSHIBA CORP**(22) Date of filing: **27.03.95**(72) Inventor: **IKEMOTO HIROYUKI****(54) PROGRAMMING SUPPORT DEVICE AND METHOD****(57) Abstract:**

**PURPOSE:** To provide a programming support device/method which can easily confirm the working states of the GUI parts.

**CONSTITUTION:** A mouse 1 and a mouse input acquisition part 2 acquire the input operations to a program using the GUI parts. A program execution part 8 and an execution memory 9 carry out and interrupt the program based on the input operations. A variable value reading/holding part 12 successively records the variable value in response to the working states of the GUI parts. Then a generation part 14 shows the working states of the GUI parts corresponding to the variable value on a display means 18 based on the recorded variable value.

COPYRIGHT: (C)1996,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-263333

(43) 公開日 平成8年(1996)10月11日

(51) Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 11/34		7313-5B	G 0 6 F 11/34	S
9/06	5 3 0		9/06	5 3 0 Q
	5 4 0			5 4 0 U

審査請求 未請求 請求項の数6 O L (全 11 頁)

(21) 出願番号 特願平7-67925

(22) 出願日 平成7年(1995)3月27日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 池本 浩幸

神奈川県川崎市幸区柳町70番地 株式会社

東芝柳町工場内

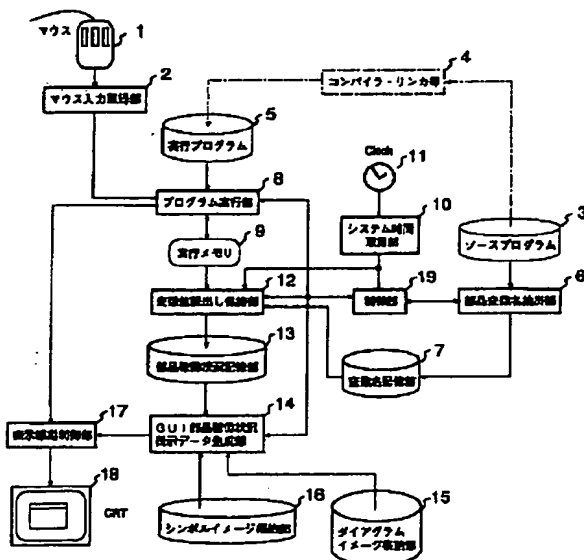
(74) 代理人 弁理士 木内 光春

(54) 【発明の名称】 プログラミング支援装置及びプログラミング支援方法

(57) 【要約】

【目的】 GUI 部品の稼働状態を容易に確認するプログラミング支援装置及びプログラミング支援方法を提案する

【構成】 マウス 1 及びマウス入力取得部 2 が、GUI 部品を用いるプログラムに対する入力操作を取得する。プログラム実行部 8 及び実行メモリ 9 が、前記プログラムを前記入力操作に基づいて実行及び中断する。変数値読出し保持部 12 が、前記 GUI 部品の稼働状況に対応する変数値を順次記録する。生成部 14 が、記録された前記変数値に基づいて、当該変数値に対応する前記 GUI 部品の稼働状況を表示手段 18 に表示する。



## 【特許請求の範囲】

【請求項 1】 GUI 部品を用いるプログラムに対する入力操作を取得する入力手段と、  
前記プログラムを前記入力操作に基づいて実行及び中断する実行手段と、  
前記 GUI 部品の稼働状況に対応する変数値を順次記録する記録手段と、  
記録された前記変数値に基づいて、当該変数値に対応する前記 GUI 部品の稼働状況を表示する表示手段と、  
を有することを特徴とするプログラミング支援装置。

【請求項 2】 前記記録する変数値の変数名として、前記プログラムから前記 GUI 部品の変数名を抽出する抽出手段を有することを特徴とする請求項 1 記載のプログラミング支援装置。

【請求項 3】 前記表示手段は、前記稼働状況を、イメージシンボルを用いたダイアグラム形式で表示するように構成されたことを特徴とする請求項 1 記載のプログラミング支援装置。

【請求項 4】 GUI 部品を用いるプログラムに対する入力操作を取得する入力のステップと、  
前記プログラムを前記入力操作に基づいて実行及び中断する実行のステップと、  
前記 GUI 部品の稼働状況に対応する変数値を順次記録する記録のステップと、  
記録された前記変数値に基づいて、当該変数値に対応する前記 GUI 部品の稼働状況を表示する表示のステップと、  
を含むことを特徴とするプログラミング支援方法。

【請求項 5】 前記記録する変数値の変数名として、前記プログラムから前記 GUI 部品の変数名を抽出する抽出のステップを含むことを特徴とする請求項 4 記載のプログラミング支援方法。

【請求項 6】 前記表示のステップでは、前記稼働状況が、イメージシンボルを用いたダイアグラム形式で表示されることを特徴とする請求項 4 記載のプログラミング支援方法。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、GUI 部品を用いるプログラムの開発を支援する、プログラミング支援装置及びプログラミング支援方法の改良に関するものである。

## 【0002】

【従来の技術】 コンピュータプログラムにおけるユーザインタフェースやその開発手法には様々なものが知られているが、そのひとつにグラフィカルユーザインタフェース (Graphical User Interface/以下「GUI」という) が存在する。GUI では、処理や機能を表すシンボルとして、アイコンやボタンのような部品 (以下「GUI 部品」という) が画面に表示され、ユーザがマウスなどポインティングデバイスのポインタでこれら GUI 部

品を操作すると、操作に応じた処理が行われる。

【0003】 GUI では、プログラムの有する様々な機能を GUI 部品として視覚的に把握できるので、プログラムの理解や操作が効率化される。そして、GUI 部品の稼働状態は、一般に、プログラムや処理の状態に応じて通常表示 (使用可能を表す)、網掛け表示 (一時的に使用不可能を表す)、非表示 (使用不可能を表す) などに变化する。

【0004】 例えば、他のプログラムを実行したり実行を中断したりするためのプログラムでは、対象となるプログラムが選択される前の状態では「実行」ボタンは網掛け表示が通常である。また、「実行」ボタンと「中断」ボタンは、プログラムが停止中か実行中かに応じていずれか一方のみを表示すべきであり、他方は無意味であるから表示する必要はない (図 6)。

【0005】 このような GUI 部品を用いるプログラムが、正しく動作しているか否かを検査したり、或いは、使い易いかどうかなどを調査する目的でプログラムの動作確認を行う作業は、プログラミング作業のかなりの比重を占めている。このような確認を行う場合、従来では、そのプログラムを実際に稼働させ、表示画面の GUI 部品を稼働状態やその変化を目視し続けることによって確認を行っていた。

## 【0006】

【発明が解決しようとする課題】 しかしながら、画面表示を目視し続けて GUI 部品の稼働状況を確認することは繁雑であり、特に、確認する利用者に継続した緊張を要求する点で精神的負担が大きかった。また、目視のみによる確認では見落としが発生する可能性を無くすることができなかった。したがって、プログラミング作業の作業効率の向上が困難であった。

【0007】 プログラミングを支援する装置としては一般にはプログラミング支援装置 (プログラミング支援方法) も知られている。しかし、従来のプログラミング支援装置では、実行プログラムによる CPU の動作をステップごとにトレースするなどとは可能であるが、GUI 部品の稼働状態をトレースすることはできなかった。

【0008】 本発明は、上記のような従来技術の問題点を解決するために提案されたもので、その目的は、GUI 部品の稼働状態を容易に確認するプログラミング支援装置及びプログラミング支援方法を提案することである。また、本発明の他の目的は、操作が容易なプログラミング支援装置及びプログラミング支援方法を提供することである。また、本発明のさらに他の目的は、GUI 部品の稼働状態を一見して容易に確認することができるプログラミング支援装置及びプログラミング支援方法を提供することである。

## 【0009】

【課題を解決するための手段】 上記の目的を達するため、請求項 1 のプログラミング支援装置は、GUI 部品

を用いるプログラムに対する入力操作を取得する入力手段と、前記プログラムを前記入力操作に基づいて実行及び中断する実行手段と、前記 G U I 部品の稼働状況に対応する変数値を順次記録する記録手段と、記録された前記変数値に基づいて、当該変数値に対応する前記 G U I 部品の稼働状況を表示する表示手段と、を有することを特徴とする。

【0010】また、請求項 2 の発明は、請求項 1 記載のプログラミング支援装置において、前記記録する変数値の変数名として、前記プログラムから前記 G U I 部品の

【0011】また、請求項 3 の発明は、請求項 1 記載のプログラミング支援装置において、前記表示手段は、前記稼働状況を、イメージシンボルを用いたダイアグラム形式で表示するように構成されたことを特徴とする。

【0012】また、請求項 4 のプログラミング支援方法は、G U I 部品を用いるプログラムに対する入力操作を取得する入力のステップと、前記プログラムを前記入力操作に基づいて実行及び中断する実行のステップと、前記 G U I 部品の稼働状況に対応する変数値を順次記録する記録のステップと、記録された前記変数値に基づいて、当該変数値に対応する前記 G U I 部品の稼働状況を表示する表示のステップと、を含むことを特徴とする。

【0013】また、請求項 5 の発明は、請求項 4 記載のプログラミング支援方法において、前記記録する変数値の変数名として、前記プログラムから前記 G U I 部品の変数名を抽出する抽出のステップを含むことを特徴とする。

【0014】また、請求項 6 の発明は、請求項 4 記載のプログラミング支援方法において、前記表示のステップでは、前記稼働状況が、イメージシンボルを用いたダイアグラム形式で表示されることを特徴とする。

【0015】

【作用】上記のような構成を有する本発明は、次のような作用を有する。すなわち、G U I 部品を用いるプログラムでは、G U I 部品の表示・非表示などの稼働状況は、必ず内部的な変数値と対応している。そして、請求項 1、4 の発明では、プログラムの実行時に、G U I 部品の稼働状況に対応する変数値が順次記録される。記録すべき変数は利用者が自ら指定してもよいし自動抽出してもよい。そして、G U I 部品の稼働状況が、前記変数値に基づいて表示されるので、稼働状況の確認はプログラム実行後に改めて行うことができる。このため、G U I 部品自体の画面表示そのものを常に目視して確認を行う必要がない。したがって、利用者の負担が軽減されるとともに、見落としの可能性も減少し、作業効率が向上する。

【0016】また、請求項 2、5 の発明では、プログラムで用いられている G U I 部品の変数名が抽出されるので、利用者がこれら変数名を予め指定しておく必要がな

く、操作が容易になる。

【0017】また、請求項 3、6 の発明では、前記稼働状況が、イメージシンボルを用いたダイアグラム形式で表示されるので、稼働状況を一見して容易に確認することができる。

【0018】

【実施例】次に、本発明の実施例について図面にしたがって具体的に説明する。なお、実施例の各機能は、所定のソフトウェアがコンピュータ及び周辺機器を制御することで実現される。すなわち、本明細書では、発明及び実施例を、各機能や各処理に対応する仮想的回路ブロック（「～手段」や「～部」など）を想定して説明している。このため、各ブロックに対して、各ハードウェア要素やソフトウェア要素は 1 対 1 には対応しない。

【0019】実施例を実現するためのコンピュータは、CPU、RAM からなる主記憶装置の他に、典型的には、キーボードやマウスなどの入力装置と、ハードディスク装置などの外部記憶装置と、CRT 表示装置やプリンタ印字装置などの出力装置と、必要な入出力制御回路を含む。但し、コンピュータの規模・構成や CPU の数・形式は自由で、例えば、コンピュータネットワーク、分散処理、マルチタスクなどを自由に用い得る。また、トラックボールやイメージスキャナなど他の入力装置や、フロッピーディスク装置・光磁気ディスク装置・フラッシュメモリなど他の記憶装置や、液晶表示装置など他の出力装置を用いてもよい。

【0020】また、実施例を実現するソフトウェアの形式は、典型的には、OS 上の応用ソフトウェアが考えられ、また、高級言語やアセンブラからコンパイルされた機械語が考えられるが、本発明が実施できる限り、OS を用いない、ソフトウェアをインタプリタで実行するなど自由に変更し得る。また、ソフトウェアの格納態様も自由で、例えば、ROM に格納しておいても、外部記憶装置から一度に又は一部づつ読み込んでも、部分ごとの格納態様が異なってもよい。また、処理の各ステップについては、その性質に反しない限り、順序の変更や並行処理は自由である。

【0021】また、本明細書において「入力」とは、外部からのみならずファイルなどからの入力を含み、本来の入力のみならずエコーバックや修正・編集などを含む。同様に、「出力」とは、外部へのみならずファイルなどへの出力を含み、本来の出力のみならず出力範囲の指定などを含む。また、ユーザインタフェースの形式も、対話形式、メニュー形式、グラフィカルユーザインタフェースやマルチウインドウなど自由である。なお、対話的入出力手順によって入力と出力を一体的操作によって実現してもよく、このような一体的操作によって、選択・指定・特定などの処理を行ってもよい。

【0022】また、本明細書におけるデータの表現形式やデータの格納手段の形式は自由で、例えば、データの

格納場所は内部記憶や外部記憶でもよく、ファイル形式を用いるかも自由である。また、データは必要な間だけ保持されれば十分でその後消滅してもよい。また、辞書データのように当面変更されない情報はROMに格納してもよい。特に、本明細書添付の図面に含まれる図形は、固定ディスク装置への限定を意味しない。

【0023】また、本明細書で明示せずとも、動作に必要な一般的な情報やその格納領域、例えば、各種ポインタ、スタック、カウンタ、フラグ、パラメータ、ワークエリア、バッファなどが適宜用いられる。

【0024】実施例の各部分が処理に要する情報は、特に記載がない場合、当該情報を保持している他の部分から獲得され、例えば、当該情報を格納している変数やメモリへのアクセスによって獲得される。また、実施例は、他のソフトウェアとともに又はその一部として実現したり、機能の一部を電子回路に置き換えてもよい。

#### 【0025】(1) 実施例の構成

本実施例は、請求項1〜6に対応するもので、本実施例の目的は、GUI部品の稼働状況を容易に確認するプログラミング支援装置及びプログラミング支援方法を提案することである。また、本実施例の他の目的は、操作が容易なプログラミング支援装置及びプログラミング支援方法を提供することである。また、本実施例のさらに他の目的は、GUI部品の稼働状況を一見して容易に確認することができるプログラミング支援装置及びプログラミング支援方法を提供することである。

【0026】まず、図1は、本実施例のプログラミング支援装置の構成を示す機能ブロック図である。すなわち、この図において、1は実行中のプログラムに対し入力操作を行うための入力手段としてのマウスであり、入力された操作は、マウス入力取得部2に入力される。これらマウス1及びマウス入力取得部2は前記入力手段を構成する。

【0027】3はプログラムの原始プログラムとしてのソースプログラムであり、多くの場合、コンパイラ・リンカ等の翻訳手段4によって、コンピュータで実行可能な実行プログラム5に変換される。なお、翻訳手段4は本発明の必須の構成要件ではなく、本装置外部で翻訳された実行プログラム5が与えられれば本発明を実施することができる。

【0028】6はソースプログラム3からGUI部品を定義している変数名を抽出する部品変数名抽出部（前記抽出手段に対応するもの）であり、抽出した変数名は、変数名記憶部7に格納される。

【0029】8は、前記入力操作に基づいて、実行メモリ9を用い、実行プログラム5を実行するプログラム実行部である。なお、GUI部品自体の表示そのものもプログラム実行部5において生成ししかるべき表示装置に表示することができる。なお、プログラム実行部8及び実行メモリ9は、前記実行手段を構成する。

【0030】10はコンピュータに内蔵された時計であるクロック11から時間情報を取得するためのシステム時間取得部であり、時間情報を提供する。

【0031】12は、抽出された前記変数名に該当する変数値を、実行メモリ9から順次読み出して部品稼働状況記憶部13に格納する変数値読出し保持部である。なお、変数名記憶部7、システム時間取得部10、クロック11、変数値読出し保持部12及び部品稼働状況記憶部13は、前記記録手段を構成する。

10 【0032】14は、記録された前記変数値を読み出し、部品の稼働状況を表示するための表示枠となる情報をダイアグラムイメージ格納部15から、部品の稼働状況を示すイメージシンボルをシンボルイメージ格納部16からそれぞれ読み出して表示データを生成するGUI部品稼働状況表示データ生成部である。また、17は、CRTなどの表示装置18に対し表示データを送出して、コンピュータの表示画面上に情報を表示するよう制御する表示駆動制御部である。

20 【0033】なお、GUI部品稼働状況表示データ生成部14、ダイアグラムイメージ格納部15、シンボルイメージ格納部16、表示駆動制御部17、表示装置18は、前記表示手段を構成する。

【0034】19は本装置各部の動作を制御する制御部であり、主に、前記時間情報に基づいて、部品変数名抽出部6、プログラム実行部8、変数値読み出し保持部12、GUI部品稼働状況表示データ生成部14などを制御する。

#### 【0035】(2) 実施例の作用及び効果

次に、具体的な画面表示やプログラムの例を用いて、本装置の動作を説明する。ここで、図2は本実施例のプログラミング支援装置の動作手順を示すフローチャートである。

30 【0036】なお、本実施例では、GUI部品を用いるプログラムを、代表的なウィンドウシステムであるXウィンドウのC言語によるプログラムとする。図3〜図5は、本実施例におけるGUI部品を用いるプログラムの一例を示す図（各々前部、中間部、後部）である。このプログラムを実行すると、利用者の操作に応じてボタンを含む画面表示が行われる。図6は、本実施例における画面表示の例を示す図である。

【0037】また、シンボルイメージ格納部16には、プログラムにおいて利用されるGUI部品の稼働状況を視覚的に表示するための、部品稼働状況に対応したイメージシンボルが予め格納されている。図7は、本実施例におけるイメージシンボルの例を示す図である。さらに、ダイアグラムイメージ格納部15には、前記イメージシンボルを表示する際の表示枠となるダイアグラムのイメージが予め格納されている。図8は、本実施例におけるダイアグラムのイメージの例を示す図である。

50 【0038】[抽出のステップ] まず、制御部19の指

示に基づいて、部品変数名抽出部6が、ソースプログラム3からGUI部品の稼働状況を表現している変数名を抽出する(ステップ201)。本実施例におけるプログラムは、上述したように、XウィンドウのC言語によるプログラムであり、変数名の取得は、ソースプログラム3の中から、Widgetという型宣言がなされた変数を検索し、その変数の名称を抽出することによって行うことができる。

【0039】次に、部品変数名抽出部6は、これら変数名を変数名記憶部7に格納する。図9は、本実施例において、変数名記憶部7に格納された変数名の例を示す図である。ここでは、図6の「実行」ボタンに対応する変数名execbtn及び「中断」ボタンに対応する変数名stopbtnなどが抽出され格納されている。このように変数名を格納した後(ステップ202)、部品変数名抽出部6は制御部19に対し処理完了を通知する。

【0040】一方、システム時間取得部10は、コンピュータに内蔵されたクロック11から現在時刻を取得し、制御部19及び変数値読み出し保持部12に現在時刻を送出している。制御部19は、部品変数名抽出部6からの処理完了の通知を受けると、システム時間取得部10から送出手続きされている現在時刻を取得し保持する(ステップ203)。

【0041】[実行のステップ]そして、制御部19の指示によって、プログラム実行部8が実行メモリ9を用いて実行プログラム5の実行を開始する(ステップ204)。プログラム5の実行では、マウス1に対する利用者の入力操作は、マウス入力取得部2を介してプログラム実行部8に送られ、プログラムの継続的な処理が行われる。プログラム実行部8は、実行プログラム5の実行中、制御部19から制御通知を待ち続け、また、実行プログラム5の実行が終了した場合は実行終了の通知を制御部19に送出する。

【0042】一方、制御部19は、プログラム実行部8から実行終了の通知を待ち続けるが、実行終了の通知がない場合、実行プログラム5の実行は継続中、すなわち、未終了とみなし(ステップ205)、実行と記録とを繰り返すようにプログラム実行部8及び変数値読み出し保持部12等を制御する。

【0043】[記録のステップ]すなわち、まず、プログラム実行部8による実行プログラム5の実行が継続中であるとき、制御部19は、システム時間取得部10から送出手続きされる現在時刻を取得し、前回取得し保持してある時刻と比較することで経過時間を算出する(ステップ206)。

【0044】そして、この経過時間が一定の時間、例えば500ミリ秒以上であれば(ステップ207)、制御部19はプログラム実行部8に対し、実行プログラム5の実行を中断するよう指示を送出する(ステップ208)。

【0045】ここで、経過時間はコンピュータの処理能力に応じて変更してよい。すなわち、この場合の経過時間は、プログラムを中断させる時間間隔を決定するが、GUI部品の稼働状況を記録する時間間隔は、処理を実行するコンピュータの処理能力によって、適切な時間間隔が異なるからである。

【0046】例えば、処理能力の高いコンピュータでは、プログラムの実行は高速で処理されるので、部品稼働状況の記録は、短い時間間隔で実施しなければならない。また、このように処理能力の高いコンピュータでは、プログラムの実行と同様に部品稼働状況を記録する処理も高速で実行されるため、部品稼働状況の記録を短い時間間隔で実施しても全体の処理が停滞するおそれがない。

【0047】逆に、処理能力の低いコンピュータを用いる場合、部品稼働状況の記録を短い時間間隔で実施すると、プログラムの処理が進まないうちに次の記録処理が行われ、結果的に全体の処理が停滞するおそれがある。このような場合は、プログラムを中断させる時間間隔である経過時間を、例えば1000ミリ秒などと、長い時間間隔とすればよい。

【0048】プログラム実行部8は、制御部19からの指示を受けると、実行プログラム5の実行を一時的に中断する(ステップ208)。続いて、制御部19は、変数値読み出し保持部12に対し実施中のGUIロプログラムの部品稼働状況を記録するよう指示を送出する。すると、変数値読み出し保持部12は、前記抽出された変数名に対応する変数値を実行メモリ9から読み出す(ステップ209)。

【0049】そして、変数値読み出し保持部12は、システム時間取得部10から送出手続きされる現在時刻を受け取り、部品稼働状況記憶部13に、記録に係る現在時刻、変数名を格納する(ステップ210)。図10は、本実施例におけるプログラムが起動してから1000ミリ秒経過後で、かつボタンが未操作の状態での部品稼働状況記憶部13の内容の例を示す図である。変数値読み出し保持部12は、部品稼働状況記憶部13への書き込みが終わると、制御部19に対し処理完了の信号を送出手続きする。

【0050】制御部19は、変数値読み出し保持部12からの処理終了の送信を受けると、保持している前回取得した時刻を、直前に取得した現在時刻で更新して保持する。そして、さらに、プログラム実行部8に対し、実行プログラム5の実行を再開するよう指示を送出する。プログラム実行部8は、実行プログラム5の処理を中断した直後の状態から続行させる(ステップ211)。

【0051】[表示のステップ]制御部19は、プログラム実行部8からの実行終了通知を待ち続け、ステップ205において、プログラム実行部8から制御部19に対し実行終了の通知があった場合は、GUI部品稼働状

況表示データ生成部 14 に対し、部品稼働状況の表示データを生成するよう指示を送出する。

【0052】制御部 19 からの指示を受けると、GUI 部品稼働状況表示データ生成部 14 は、まず、部品稼働状況記憶部 13 から、格納されている時刻、変数名、変数値を読み出す（ステップ 212）。続いて、生成部 14 は、イメージシンボルと（ステップ 213）、表示枠となる表示データを読み出す（ステップ 214）。そして、生成部 14 は、表示枠内に、時刻、部品の変数名とともに、変数値の変化に対応するイメージシンボルを表示するためのデータを生成し、この表示データを表示駆動制御部 17 に送出する。

【0053】表示駆動制御部 17 は、この表示データを CRT などの表示装置 18 に表示するよう制御を行い、表示装置 18 に GUI 部品の稼働状況が表示される（ステップ 215）。図 11 は、本実施例における GUI 部品稼働状況の画面表示の一例を示す図である。

【0054】この表示では、実行ボタン(execbtn)と中断ボタン(stopbtn)が同時に表示されることがなく、且つ、必然的に実行ボタンと中断ボタンとが同時に入力可能とならないことを容易に確認することができる。

【0055】このように、本実施例によれば、稼働状況の確認はプログラム実行後に改めて行うことができる。このため、GUI 部品自体の画面表示そのものを常に目視して確認を行う必要がない。したがって、利用者の負担が軽減されるとともに、見落としの可能性も減少し、作業効率が向上する。

【0056】また、本実施例では、プログラムで用いられている GUI 部品の変数名が抽出されるので、利用者がこれら変数名を予め指定しておく必要がなく、操作が容易になる。さらに、本実施例では、前記稼働状況が、イメージシンボルを用いたダイアグラム形式で表示されるので、稼働状況を一見して容易に確認することができる。

#### 【0057】(3) 他の実施例

なお、本発明は上記実施例に限定されるものではなく、実施態様の変更は自由であるから、次に例示するような他の実施例をも包含するものである。例えば、プログラム言語や GUI 部品の内容は自由であり、ウィンドウシステムは必須ではない。また、実行対象となるプログラムの種類も、機械語のロードモジュール、インタプリタ言語、中間コードなど自由である。

【0058】また、記録する変数値の変数名は自動抽出せず、利用者が自由に設定してもよい。また、GUI 部品の稼働状況はシンボルイメージを用いたダイアグラム形式には限定されず、文字列で表示するなど自由な形式で表示してもよい。また、表示は画面表示装置のみならずプリンタ出力などによって行ってもよい。

#### 【0059】

【発明の効果】以上説明したように、本発明によれば、GUI 部品の表示結果を容易に確認できるので、プログラミングの効率が向上する。

#### 【図面の簡単な説明】

【図 1】本発明の実施例のプログラミング支援装置の構成を示す機能ブロック図

【図 2】本発明の実施例のプログラミング支援装置の動作手順を示すフローチャート

10 【図 3】本発明の実施例における GUI 部品を用いるプログラムの一例を示す図（前部）

【図 4】本発明の実施例における GUI 部品を用いるプログラムの一例を示す図（中間部）

【図 5】本発明の実施例における GUI 部品を用いるプログラムの一例を示す図（後部）

【図 6】本発明の実施例における画面表示の例を示す図

【図 7】本発明の実施例におけるイメージシンボルの例を示す図

20 【図 8】本発明の実施例におけるダイアグラムのイメージの例を示す図

【図 9】本発明の実施例において、変数名記憶部 7 に格納された変数名の例を示す図

【図 10】本発明の実施例におけるプログラムが起動してから 1000 ミリ秒経過後で、かつボタンが未操作の状態での部品稼働状況記憶部 13 の内容の例を示す図

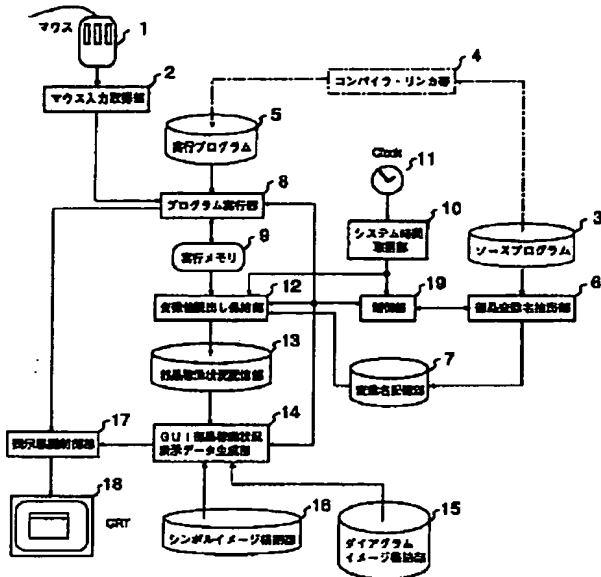
【図 11】本発明の実施例における GUI 部品稼働状況の画面表示の一例を示す図

#### 【符号の説明】

- 1…マウス
- 30 2…マウス入力取得部
- 3…ソースプログラム
- 4…翻訳手段
- 5…実行プログラム
- 6…部品変数名抽出部
- 7…変数名記憶部
- 8…プログラム実行部
- 9…実行メモリ
- 10…システム時間取得部
- 11…クロック
- 40 12…変数値読み出し保持部
- 13…部品稼働状況記憶部
- 14…GUI 部品稼働状況表示データ生成部
- 15…ダイアグラムイメージ格納部
- 16…シンボルイメージ格納部
- 17…表示駆動制御部
- 18…表示装置
- 19…制御部
- 200以降…手順の各ステップ



【図1】



【図7】

GUI部品種別状況	シンボル
非表示状態	W
通常表示状態	AA
表示網掛け状態	—

【図9】

oplevel
rc
execbtn
stopbtn
quitbtn

【図5】

```

XtSetArg(args[0], XmNx, 70);          XtSetArg(args[1], XmNy, 1);
XtSetArg(args[2], XmNwidth, 60);      XtSetArg(args[3], XmNheight, 30);
stopbtn = XmCreatePushButtonGadget(rc, "中断", args, 4);
XtAddCallback(stopbtn, XmNactivateCallback, ButtonCB, "2");
XtManageChild(stopbtn);

```

```

XtSetArg(args[0], XmNx, 140);          XtSetArg(args[1], XmNy, 1);
XtSetArg(args[2], XmNwidth, 60);      XtSetArg(args[3], XmNheight, 30);
quitbtn = XmCreatePushButtonGadget(rc, "終了", args, 4);
XtAddCallback(quitbtn, XmNactivateCallback, ButtonCB, "3");
XtManageChild(quitbtn);

```

```

XtUnmanageChild(stopbtn);
XtRealizeWidget(oplevel);
XtAppMainLoop(app_context);

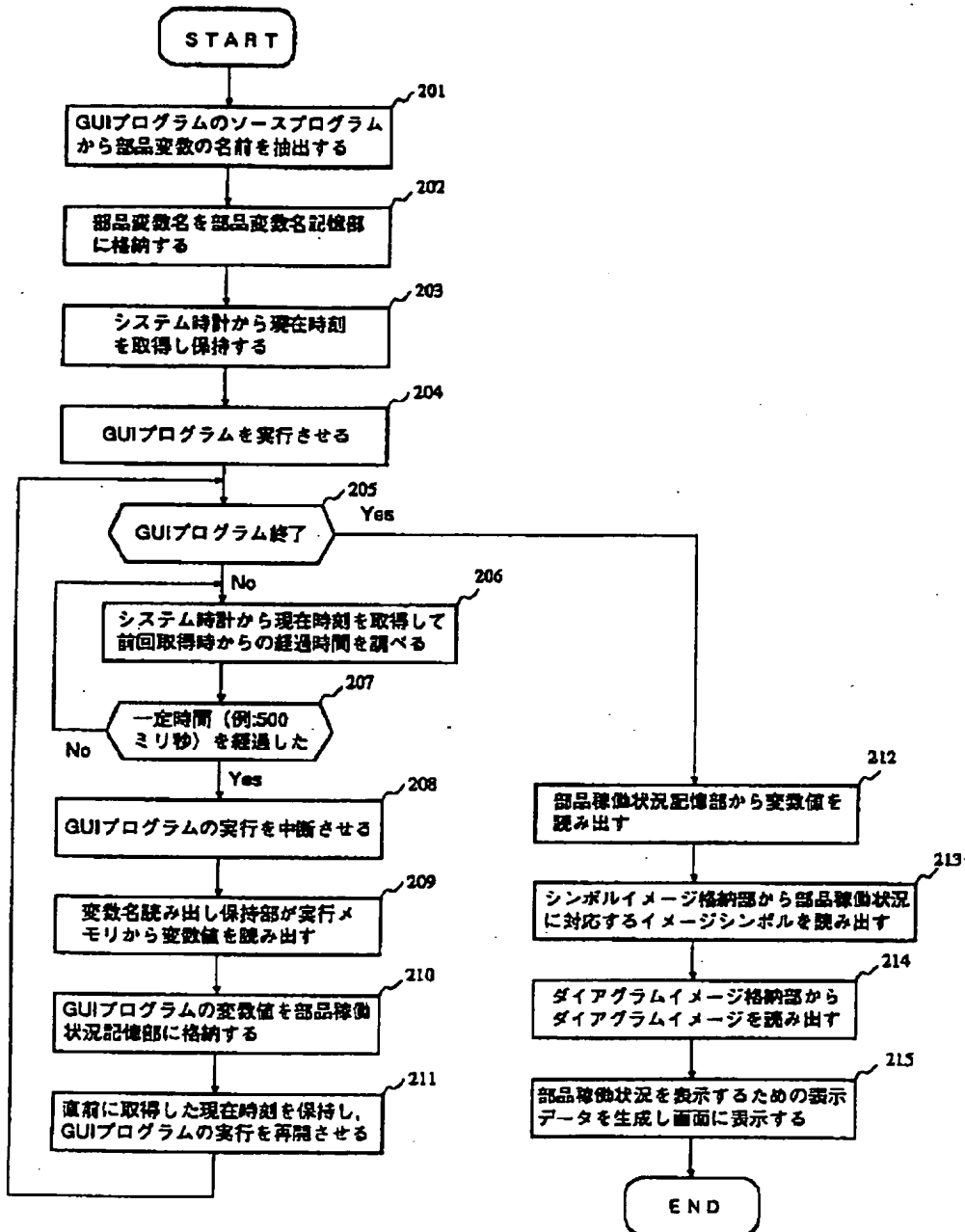
```

```

}

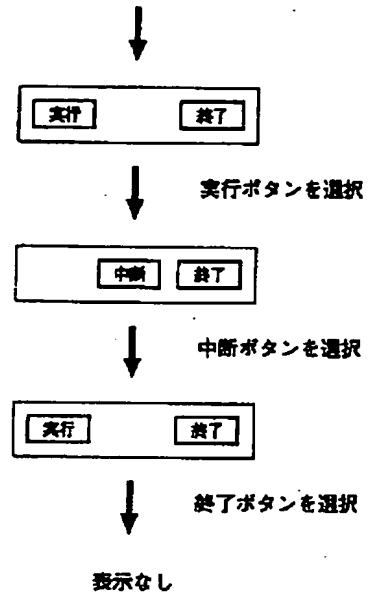
```

【図 2】



【図6】

## GUIプログラムを起動



【図 8】

[illegible]

【図4】

```

void ExecStop()
{
    Arg args[5];
    XtAppContext app_context;
    XmString string;

    XmJpInitialize(&argc, argv, 0);
    toplevel = XtAppInitialize (&app_context, "PopupMenu", NULL, 0, &argc, argv, NULL, FULL, 0);

    XtSetArg(args[0], XmNwidth, 150); XtSetArg(args[1], XmNheight, 50);
    XtSetArg(args[2], XmNresizeWidth, False);
    XtSetArg(args[3], XmNresizeHeight, False);
    XtSetArg(args[4], XmNadjustLast, False);
    rc = XmCreateBulletinBoard(toplevel, "rc", args, 5);
    XtManageChild(rc);

    XtSetArg(args[0], XmNx, 1); XtSetArg(args[1], XmNy, 1);
    XtSetArg(args[2], XmNwidth, 60); XtSetArg(args[3], XmNheight, 30);
    execbtn = XmCreatePushButtonGadget(rc, "実行", args, 4);
    XtAddCallback(execbtn, XmNactivateCallback, ButtonCB, "1");
    XtManageChild(execbtn);
}

```

【図 10】

時刻	toplevel	rc	execbin	stopbin	quitbin			
500	Y	Y	Y	Y	Y			
1000	NA	NA	NA	Y	NA			

通常表示を示す値                      非表示を示す値

【図 1 1】

[illegible]